IN THE UNITED STATES DISTRICT COURT
FOR THE WESTERN DISTRICT OF TEXAS
MIDLAND-ODESSA DIVISION

MALIKIE INNOVATIONS LTD., KEY PATENT INNOVATIONS LTD. Plaintiffs, v. MARA HOLDINGS, INC. (F/K/A MARATHON DIGITAL HOLDINGS, INC.) Defendant.	CASE NO. 7:25-CV-00222-DC-DTG JURY TRIAL DEMANDED

PLAINTIFFS' RESPONSIVE CLAIM CONSTRUCTION BRIEF

TABLE OF CONTENTS

TABLE OF AUTHORITIES

Page(s)

Cases

I. INTRODUCTION

Plaintiffs Malikie Innovations Ltd. and Key Patent Innovations Ltd. ("Plaintiffs") submit their Responsive Claim Construction Brief regarding disputed terms in U.S. Patent Nos. 8,532,286 ("'286 Patent"), 7,372,960 ("'960 Patent"), 8,666,062 ("'062 Patent"), 8,788,827 ("'827 Patent"), 10,284,370 ("'370 Patent"), and 7,372,961 ("'961 Patent") (collectively, the "Asserted Patents").

The Asserted Patents relate to innovations in the field of cryptography. Their inventors were technologists at Certicom Corporation and BlackBerry Limited (which acquired Certicom). Certicom was a pioneer in elliptic curve cryptography (ECC), a way of performing public key cryptography as an alternative to the older and less secure RSA technique. Certicom received numerous patents for its innovations in this field, the Asserted Patents among them.

In brief, the '286 Patent relates to techniques for reducing computation and storage requirements of performing Montgomery style reduction (an operation used in cryptographic operations). The '062 and '960 Patents share a common specification and relate to finite field engines (computer software and/or hardware for performing finite field operations) for use with cryptographic systems. The '827 and '370 Patents, which also share a common specification, relate to techniques for accelerating the verification of ECC digital signatures. The '961 Patent relates to secure key generation techniques for public key cryptosystems.

Plaintiffs' proposed construction for each of the terms at issue reflects the plain and ordinary meaning, as a person of ordinary skill in the art (POSITA)[1] would have understood them at the time of invention, in view of each patents' disclosures and prosecution histories.

[1] Plaintiffs' technical expert, Dr. Paul Martin, opines that a POSITA would have possessed (1) a bachelor's degree in computer science, mathematics, applied mathematics, or a related field, and two or more years of experience in applied cryptography or computer security software engineering, or (2) an advanced degree in computer science or a related field and or more years of experience in applied cryptography or computer security software engineering. Ex. 1 ¶ 41.

II. ARGUMENT

A. The '286 Patent

The '286 patent (Dkt. 52-3) is entitled "System and Method for Reducing the Computation and Storage Requirements for a Montgomery-Style Reduction" and relates generally to "an alternative way in which to produce a Montgomery reduction," a common use method. '286 Pat., Title; Abstract.

In public key cryptography, certain operations use modular arithmetic, including modular reduction. '286 Pat., 1:20-30. "For example, to multiply two numbers modulo some [number] n, the classical approach is to first perform the multiplication and then calculate the remainder."[2] *Id*. "The calculation of the remainder is referred to as reduction," and it is considered to be a slow process. *Id*. One well known method for modular reduction is Montgomery modular reduction, often referred to as "Montgomery reduction." *Id.* at 1:31-35. The '286 patent relates generally to "an alternative way in which to produce a Montgomery reduction." *Id*. Abstract.

Conventional Montgomery reduction reduces a value "a" using a Montgomery radix "R" to calculate aR^{-1} mod n. *Id*. at 1:47-64. A precomputed value "μ" is used to calculate a multiplier "$m=\mu a \bmod 2^{w}$." A multiple of the modulus "n" (*i.e.*, m×n) is then added to "a" (*i.e.*, a+m×n is computed) to "zero" out the least significant word (LSW) of "a," and the result is "shifted down" to eliminate that LSW.[3] *Id*. at 2:47-61, 5:4-17. Adding the multiple of the modulus to "a" to

[2] The "modulo" operation calculates the remainder of dividing one number by another called the "modulus." Ex. 1 ¶ 61. For example, 14 mod 12 = 2 (*i.e.*, 14 ÷ 12 = 1 remainder 2).

[3] A "shift" is a programming operation that moves the binary digits (bits) of a number to the left or right, which is equivalent to multiplying or dividing the number by a power of two. For example, 00010101 shifted to the right by two bits is 00000101 and is equivalent to dividing by 2^2 or multiplying by 2^{-2}. Ex. 1 ¶ 62.

eliminate its least significant word "cancels" it. *Id*. Repeating this process effects the reduction aR^{-1} mod n. *Id*. The values μ and n are stored in registers throughout this process.[4] *Id*.

To reduce the number of stored values and computations in accordance with the invention, the '286 Patent teaches the use of "a modified reduction value" that "can be used in place of μ and n." *Id*. at 5:28-36. A "logical shift" or a "signed version" of the reduction value may alternatively be used. *Id*. Rather than being cancelled using a multiple of the modulus, the LSW is "replaced" with a different value generated using the reduction value. *Id*. at 5:45-67; Fig. 6. Replacement using the modified reduction value obviates the need to multiply "a" by μ and determine m. *Id*. "This … avoids both the multiplication necessary to compute m and the storage required for μ." *Id*. at 6:2-9. It also avoids the need to store n. *Id*.

1. "Montgomery-style reduction" *('286 patent - claims 1, 5, 6, 9)*

Defendant's Construction	Plaintiffs' Construction
"reduction that proceeds by clearing the least significant portions of an unreduced operand and leaving the remainder in the more significant portions"	This term appears only in the preamble and is not limiting.

The Federal Circuit "recognize[s] that as a general rule preamble language is not treated as limiting." *Arctic Cat Inc. v. GEP Power Prods., Inc*., 919 F.3d 1320, 1327 (2019) (quoting *Aspex Eyewear, Inc. v. Marchon Eyewear, Inc.*, 672 F.3d 1335, 1347 (Fed. Cir. 2012) (internal citations omitted); *see also Am. Med. Sys., Inc. v. Biolitec, Inc*., 618 F.3d 1354, 1358 (2010). A preamble may limit an invention only "if it recites essential structure or steps, or if it is 'necessary to give life, meaning, and vitality' to the claim." *Catalina Mktg. Int'l, Inc. v. Coolsavings.com, Inc.*, 289 F.3d 801, 808 (Fed. Cir. 2002) (quoting *Pitney Bowes, Inc. v. Hewlett-Packard Co.*, 182 F.3d 1298,

[4] A register is hardware in a CPU for temporary storage of data during program execution. Ex. 1 ¶ 62.

1305 (Fed. Cir. 1999)). Factors relevant to determining whether a preamble is limiting include whether the preamble (1) provides antecedent basis for terms in the body of the claim; (2) supplies "structure needed to make the body itself a 'structurally complete invention'"; and (3) was "relied on during prosecution to distinguish prior art." *Arctic Cat*, 919 F.3d at 1329. None of these factors are present here, thus confirming that the term "Montgomery-style reduction," which appears only in the preamble, is not limiting.

Claim 1 recites:

> **1.** A method for performing, on a cryptographic apparatus, a ***Montgomery-style reduction*** in a cryptographic operation, the method comprising:
>
> obtaining an operand for the cryptographic operation;
>
> computing a modified operand using a reduction value, instead of a modulus used in performing a standard Montgomery reduction, to perform a replacement of a least significant word of the operand, rather than perform a cancellation thereof, the reduction value being a function of the modulus; and
>
> outputting the modified operand.

The term "Montgomery-style reduction" appears only in the preamble and does not provide antecedent basis to any term in the body of the claim. Rather, the term is used merely to describe and state the purpose of the invention embodied by the steps of the claim. *Catalina*, 289 F.3d at 808 ("[A] preamble is not limiting where a patentee defines a structurally complete invention in the claim body and uses the preamble only to state a purpose or intended use for the invention." (cleaned up)); *see also Am. Med. Sys.*, 618 F.3d at 1359. The body of the claim itself recites the necessary steps ("obtaining," "computing," and "outputting") to perform the claimed method, making the preamble unnecessary to "the structure or steps of the claimed invention." *Id.* at 1358-59. Moreover, "Montgomery-style reduction" was not relied on during prosecution to distinguish the invention from prior art; the applicant relied only on features in the body of the claim. *See* Ex. 2 at 5592-93, 5624-26; *Intirtool, Ltd. v. Texar Corp.*, 369 F.3d 1289, 1295 (Fed. Cir. 2004).

Defendant offers no meaningful evidence in support of its attempt to elevate the term to limiting status. Defendant's specification citations relating to "provid[ing] an alternative way in which to produce a Montgomery reduction" and "modifying the Montgomery reduction mechanism" (Dkt. 52 at 10) do nothing to justify transforming the "Montgomery-style reduction" descriptor into a limitation. Instead, they underscore the fact that the body of the claim itself, *i.e.,* the elements of the claim (without any need of the preamble) is what defines the particular "alternative way" or "modification" of reduction method claimed in Claim 1. For example, the specification teaches:

> an alternative way in which to produce a Montgomery reduction from below [is] by storing a new precomputed value used to substantially replace the μ and n values used in Montgomery reduction with a single value. This may be done by storing a **modified reduction value** in the cryptographic apparatus, wherein the **modified reduction value**, **when applied to an operand**, input to or generated by, the cryptographic apparatus, **performs a replacement for values** in a low-order segment which is a target of the reduction, **rather than a cancellation thereof, as performed in a standard Montgomery reduction**; and performing the reduction from below using the modified reduction value.
>
> By modifying the Montgomery reduction mechanism in this way, the number of multiplications and registers required to effect the Montgomery reduction can be reduced.

'286 Pat., 3:27-45 (emphasis added). This is one example of an "alternative way" of conducting a Montgomery reduction that does not require additionally defining what "Montgomery-style reduction" is. And it confirms that rather than being part of the invention, the term "Montgomery-style reduction" merely describes the purpose or use of the invention set forth in the body.

The Federal Circuit's decision in *Arctic Cat* is instructive. There, the Court affirmed, on *de novo* review, the lower court's finding that the preamble, "A personal recreational vehicle comprising," was not limiting, even though "the structure is 'underscored as important by the specification.'" *Arctic Cat*, 919 F.3d at 1323, 1329 (quoting *Catalina*, 289 F.3d at 808). The Court noted that the preamble lacked characteristics often used to identify whether a preamble "*is*

limiting," including whether it provides antecedent basis, contains necessary structure or is relied on to distinguish prior art. *Id.* at 1329-30. The Court also noted that the "vehicle in the preamble is entirely conventional apart from the improvement [claimed] in the body of the claims." *Id.* So too here. The novelty of the invention of Claim 1 is captured in the recited elements of the claim. As the specification explains (*supra*), the claimed method is an "alternative way" or "modification" of conventional Montgomery-style reduction.

Defendant's cases are inapposite. The preamble in *Corning Glass Works v. Sumitomo Elec. U.S.A, Inc.*—"An optical waveguide comprising"—does not include a statement of purpose or intended use and defines type of structure claimed, which the court found was not sufficiently described in the body of the claim. 868 F.2d 1251, 1256-57 (Fed. Cir. 1989). Here, the preamble employs the standard pattern of intended use language, and the body describes a complete invention. *See supra*. Similarly, the preamble in *Gen. Elec. Co. v. Nintendo Co.* does not "merely" state the purpose or intended use of "[a] system for displaying a pattern on a raster scanned display device," but further recited *how* the system does it, *i.e.*, "by mapping bits … onto the raster," which "restricted [the claim] to those display devices that work by displaying bits." 179 F.3d 1350, 1361-62 (Fed. Cir. 1999). That is not the case here, where the preamble does no work other than to state a purpose or intended use of the method.

Defendant also argues that the preamble must be limiting to avoid alleged prior art ("such as the prior art pseudo-Mersenne"), Dkt. 52 at 11, but this impermissibly "put[s] the validity cart before the claim construction horse." *Landers v. Sideways, LLC*, 142 F. App'x 462, 468 (Fed. Cir. 2005) (citing *Nazomi Commc'ns, Inc. v. ARM Holdings, PLC*, 403 F.3d 1364, 1367-69 (Fed. Cir. 2005)). The focus of claim construction is not on validity, but on understanding the claims in light of the intrinsic record. *Id*. The intrinsic record shows the preamble is not limiting. Defendant's

attempt to litigate a new invalidity theory during claim construction should be ignored.[5] *See Sesaco Corp. v. Equinom Ltd.*, No. 1:20-CV-01053-LY, 2022 WL 17257244, at *6 (W.D. Tex. Nov. 28, 2022) (validity issue ancillary to claim construction "a question for another day").

If the Court finds the preamble limiting, it should reject Defendant's version of the specification's description of Montgomery reduction. The specification states "Montgomery reduction reduces from below, that is, the method proceeds by clearing the least-significant portions of the unreduced ***quantity***, leaving the remainder in the upper portion." '268 Pat., 1:41-46 (emphasis added). Defendant's construction adopts the description except for the word "quantity," which it replaces with "operand." Defendant contends "operand," as claimed in the body of the claim, is an "unreduced quantity," making it appropriate to replace "quantity" with "operand." Dkt. 52 at 11. This is wrong for at least two reasons. First, Defendant's word-swap reads out preferred embodiments. The specification teaches the invention can be performed in multiple iterations of an operation, which means it can be performed on a previously or partially reduced operand. Ex. 1 ¶¶ 64-67; *see* '286 Pat., 5:45-49 (modified reduction value used "at each iteration"); 5:12-19 (obtaining and using modified reduction value "for iterations 1 to k-1"); 6:32-35 (obtaining and using modified reduction value "at each iteration"); 6:40-44 ("This process is repeated k-1 [times] …."). Defendant's construction—which requires operating on an *unreduced* operand—would read out these embodiments. *Kaneka Corp. v. Xiamen Kingdomway Grp. Co.*, 790 F.3d 1298, 1304 (Fed. Cir. 2015) ("A claim construction that excludes a preferred embodiment is rarely, if ever, correct." (cleaned up)). Second, use of "an unreduced operand" in the preamble would create unnecessary confusion—the "unreduced operand" of the preamble would ***not*** be the

[5] Defendant relies on its expert's analysis of a reference ("Guajardo") not cited in Defendant's invalidity contentions. Ex. 3 at 5-10.

antecedent basis for the "operand" of the claim, notwithstanding Defendant's argument that it would be "consistent with the claim language." Using "quantity" is both consistent with the broader scope intended by the specification and avoids confusion.

2. "perform a replacement of a least significant word of the operand"
('286 patent - claims 1, 5, 6, 9)

Defendant's Construction	Plaintiffs' Construction
"add a modular equivalent of the operand's least significant word to the more significant words of the operand such that the result can be shifted down to drop the least significant word"	"replace a word that makes the smallest contribution to the value of the operand"

The plain and ordinary meaning of "to perform a replacement of a least significant word of the operand," as it would have been understood by a POSITA in view of the intrinsic record, is to "replace a word that makes the smallest contribution to the value of the operand." This construction is straightforward and aligns with the claim language and the patent specification.

Plaintiffs' proposed construction defines what it means to replace "a least significant word" ("LSW") of the operand. Defendant agrees that a LSW is "a word that makes the smallest contribution to the value," Dkt. 52 at 11-12 ("The parties agree that 'least significant word of the operand' from the claim term is a word that makes the smallest contribution to the value of the operand …"), but Defendant does not incorporate this meaning into its construction. Given the parties' agreement, Plaintiffs' construction should be adopted at least as to this term.

Defendant contends that "to perform a replacement" requires construction beyond the term's plain meaning, which is to "replace" (as reflected in Plaintiffs' proposal). But nothing in the intrinsic record compels a construction different from this plain meaning. And nothing supports the wholesale recasting of this ordinary term in the way Defendant urges.

The invention involves "obtaining an operand" and "computing a modified operand using

a reduction value … to ***perform a replacement*** of a least significant word of the operand." '286 Pat., Claim 1. The specification confirms that "replacement" is used in its ordinary sense. Ex. 1 ¶¶ 68-71. For example, it teaches that a "modified reduction value" may be used to "***perform[] a replacement for values***." *Id*. at 3:27-39. The specification describes an example where a reduction value (n') is used to perform a replacement of the least significant word (a_0) of an operand ($a \equiv [\ldots, a_4, a_3, a_2, a_1, a_0]$) where "the value a_0 ***can be replaced with*** $a_0 \times n' \times 2^w$." *Id*. at 5:59-60. In other words, the least significant word of the operand is "replaced" with something else, *i.e.*, the ordinary meaning of "replacement." Ex. 1 ¶¶ 68-71. The patent further teaches that "using the modified reduction value n' … the least significant word ***is removed***" and a different value is added back to the remaining words. *Id*. at 6:12-19; Fig. 6. This also describes "replacement" in the ordinary sense.[6] Ex. 1 ¶ 70. Because the patentee did not "clearly set forth a definition of ["replacement"] other than its plain and ordinary meaning" or "clearly express an intent to redefine [it]," the Court should adopt Plaintiffs' construction and reject Defendant's. *Hill-Rom Servs., Inc. v. Stryker Corp.*, 755 F.3d 1367, 1371 (Fed. Cir. 2014).

Defendant's construction is incorrect because it deviates from plain meaning and improperly limits Claim 1 to a specific embodiment in the specification. Defendant's construction requires "add[ing] a modular equivalent of the operand's least significant word." Defendant's only support for injecting this limitation is the value $a_0 \times n' \times 2^w$ in the embodiment described at 5:59-67 and 6:32-38. Dkt. 52 at 13 ("adding a modular equivalent of a_0 (i.e., $\mathbf{a_0 \times n' \times 2^w}$) to the remaining words"); Ex. 4 (Koç Dep. Tr.) at 56:5-58:9 (confirming reliance on $a_0 \times n' \times 2^w$). The specification

[6] The prosecution history is in accord. For example, in Office Actions dated 7/10/2012 and 3/22/2013, the PTO interpreted "perform a replacement" to read on a reference that allegedly taught "the value of the least significant word ***replaced with*** the output of [an] operation." Ex. 5 at 5561, 5607.

explains that n'×2^w is a "shifted" version of the reduction value n'. '286 Pat., 5:56-59 ("if the value n' is then ***shifted up*** by one digit, which is equivalent to multiplying by 2^w, a value is obtained that is equivalent to 1 mod n"), 5:59-60 ("the value a_0 can be replaced with a_0×n'×2^w, that is, a_0 multiplied by n' ***shifted up*** one digit"), 5:62-67 ("a_0×n'×2^w, taken without reduction, is zero in its least significant digit (by ***the shift 2^w***)"). But claim 1 is not limited to using a shifted version of the reduction value. *Id*., Claim 1 ("using a reduction value"); *see id*. at 5:31-35 ("a modified reduction value or a logical shift or signed version of such a value can be used"); Ex. 1 ¶¶ 72-75. In fact, dependent Claim 2 specifies that "the reduction value is n'=2^{-w}mod n, ***or*** a shifted or signed version of n'." *Id*., Claim 2. Defendant identifies no "words or expressions of manifest exclusion or restriction" that justifies its restrictive construction. *Hill-Rom Services*, 755 F.3d at 1371-72 (quoting *Liebel–Flarsheim Co. v. Medrad, Inc.*, 358 F.3d 898, 906 (Fed.Cir.2004)).

3. "perform a cancellation thereof" *('286 patent - claims 1, 5, 6, 9)*

Defendant's Construction	Plaintiffs' Construction
"add a multiple of the modulus to the operand such that the least significant word of the result is zero and the result can be shifted down to drop the least significant word"	"add a multiple of the modulus to the operand to eliminate the least significant word of the operand"

Both parties agree that "cancelling," in the context of the record, requires "adding a multiple of the modulus to the operand." Plaintiffs' construction also makes clear that performing a cancellation results in "eliminating the least significant word of the operand," *i.e.,* the ordinary meaning of "cancelling" is "eliminating."

The '286 Patent uses the term "cancellation" once in the specification and then again only in the claims. '286 Pat., 3:31-39. "Cancellation" does not have a particular meaning in the field of art of the invention beyond its ordinary meaning. Ex. 1 ¶ 79. The '286 Patent uses the term in describing an improved technique of "using a reduction value, ***instead of a modulus used in***

performing a standard Montgomery reduction, ***to perform a replacement of a least significant word of the operand***, ***rather than perform a cancellation thereof***." '286 Pat., claim 1; *see id*. at 3:31-39 (similar). In standard Montgomery reduction, a multiple (m) of the modulus (n) is added to the operand (a) — *i.e.*, a+m×n is computed — to "zero" the least significant word of the operand, and the operand is "shifted down," thereby eliminating the least significant word. *Id*. at 4:65-5:23; *see also* 2:47-53, 4:40-47; Fig. 4; Ex. 1 ¶¶ 77-78. Because fewer words remain in "a" after each iteration, a+m×n is computed each time to eliminate the next least significant word. *Id*. Accordingly, cancellation is used in its ordinary sense, *i.e.*, to eliminate something. Ex. 1 ¶ 76-78.

Defendant's construction approximates a similar meaning but errs in meaningful ways. First, Defendant says "the least significant word of the ***result***" (as opposed to "operand") – this injects unnecessary new terminology; the portion of the specification Defendant cites from makes clear that "a" is the operand to which the multiple of the modulus (mn) is added. Second, Defendant says "the result ***can be shifted down*** to drop the least significant word." The specification explains that cancellation *necessarily* requires shifting down in order to eliminate (or cancel) the least significant word. Ex. 1 ¶ 77. Defendant's permissive phrasing of "can be" leaves room for shifting not to occur, which is antithetical to the concept of canceling.

B. The '062 and '960 Patents

Cryptographic protocols often require the use of "keys." '960 Pat., 1:16-55. In elliptic curve cryptography (ECC), an elliptic curve is specified with "a finite field and an equation over that finite field," where "[t]he points on the elliptic curve are the pairs of finite field elements satisfying the equation of the curve." *Id*. To perform calculations involving points on the elliptic curve, "calculations are done in the underlying finite field." *Id*.

The strength of an ECC system depends on the key size; larger keys provide more security. *Id*. at 1:66-2:7. "[H]owever, different key sizes require defining different elliptic curves over

different finite fields," where, in general, the size of the finite field increases with cryptographic strength. *Id*. Accordingly, multiple finite fields may need to be supported. *Id*. at 2:8-24. Using "specific methods for each finite field leads to more efficient code since it may be optimized to take advantage of the specific finite field," but "will increase the code size dramatically." *Id*. Conversely, using "generic method[s] prevents the use of optimization techniques" which makes the code smaller but less efficient. *Id*. For example, because finite field elements are often too long to fit into one machine word (requiring programs to deal with multiple words)[7], either efficient code tailored to the number of words that must be dealt with can be used, or smaller (but slower) wordsize non-specific code can be used. *Id*. at 2:25-62.

To enable fast engines to be produced for specific finite fields without duplicating the bulk of instructions, *id*. at 4:10-27, the patents teach first performing a "wordsized" finite field operation on "wordsized" representations of finite field elements (producing an unreduced result), followed by a "specific" modular reduction "corresponding to the particular finite field identified" (producing a reduced result). *Id*. at 8:26-34, 8:40-60. The specific modular reduction should also "lower the length of the result to the appropriate word length of the underlying finite field." *Id*.

1. "finite field operation" *('960 patent - claims 3, 6; '062 patent - claims 1-4, 6, 7)*[8]

Defendant's Construction	Plaintiffs' Construction
"operation where each operand is a finite field element"	"operation in a finite field"

Claim 1 of the '062 Patent (Dkt. 52-5) and claim 3 of the '960 Patent (Dkt. 52-4) each

[7] A machine word is a unit of data (in bits) that a computer processor can handle in a single operation. Ex. 1 ¶ 81.

[8] The '960 and '062 Patents share a common specification; for convenience, citations made to the '960 patent are also applicable to the '062 patent.

recite a method of "performing a finite field operation on elements of a finite field"[9]:

> **('062 patent) 1.** A method of performing a ***finite field operation*** on elements of a finite field, the method comprising a processor:
>
> obtaining a first set of instructions for performing the ***finite field operation*** on values representing the elements of the finite field;
>
> executing the first set of instructions to generate an unreduced result completing the ***finite field operation***;
>
> ...
>
> **('960 patent) 3.** A method of performing a ***finite field operation*** on elements of a finite field, comprising the steps of:
>
> …
>
> performing a non-reducing wordsized operation on said representations, said wordsized operation corresponding to said ***finite field operation***;
>
> …

The plain and ordinary meaning of "finite field operation," as used in the claims and as that term would have been understood by a POSITA at the time of the invention, is "operation in a finite field." Ex. 1 ¶¶ 83-85. Plaintiffs' construction "stays true to the claim language and most naturally aligns with the patent's description of the invention." *Phillips v. AWH Corp.*, 415 F.3d 1303, 1316 (Fed. Cir. 2005) (attribution omitted); *see also* '960 Pat., 4:10-12 ("In general terms, the invention provides … methods for operating on elements in a finite field."); *see id*. at Abstract.

Defendant's construction, on the other hand, includes the phrase "each operand is a finite field element." This construction improperly narrows the scope of the claim and would read out preferred embodiments. The '960 and '062 patents teach that finite field operations are used to operate on elliptic curve points, for example, as part of an elliptic curve operation. *Id*., 7:25-29 ("Each elliptic curve operation 320 requires certain finite field operations, and so accordingly pointers 330 are provided to operations in the finite field engine 400 …."); 7:48-51 ("The finite

[9] Defendant agrees that this term should be construed in a manner consistent with its use in the claims' preambles. Dkt. 52 at 16.

field elements [comprising an elliptic curve point] are … operated on directly by the finite field engine 400 ….”), 8:5-10 (elliptic curve operations “in turn direct finite field operations”); 8:46-48 (“The finite field engine 400 provides finite field routines 430 for use by … the elliptic curve engine 300.”). The specification also makes clear that an elliptic curve point consists of ***pairs*** of finite field elements. ’960 Pat., 1:45-47 (“The points on the elliptic curve are the ***pairs*** of finite field elements satisfying the equation of the curve ….”); 7:46-48 (“an elliptic curve point consists of ***two*** finite field elements”); Dkt. 52-2 (Koç Decl.) ¶ 68 (acknowledging that “elliptic curves points … each consist of ***two*** finite field elements as the *x*-coordinate and *y*-coordinate of the point”). And, while the patent teaches in one embodiment that finite field elements “are only operated on ***directly*** by the finite field engine,” the POSITA would understand that they are operated on at least indirectly by elliptic curve operations because they “require,” “direct,” and “use” finite field operations. ’960 Pat. 7:25-29, 7:46-52, 8:5-10, 8:46-48; Ex. 1 ¶ 85. The patent is therefore clear that finite field operations can be used to operate on elliptic curve points, which are pairs of finite field elements. *Id*.

Defendant’s construction limits “finite field operations” only to those where “***each*** operand is ***a*** finite field element,” *i.e.*, one finite field element per operand. This would forbid finite field operations on ***pairs*** of finite field elements, such as those comprising elliptic curve points, in direct contradiction to embodiments in the specification. Nothing in the claim language requires “finite field operation” to be so limited. Indeed, both claims claim “performing a finite field operation on ***elements*** of a finite field.” Defendant’s construction would also impermissibly read out the preferred embodiments noted above. *Kaneka Corp.*, 790 F.3d at 1304 (“A claim construction that excludes a preferred embodiment is rarely, if ever, correct.” (cleaned up)).

Defendant’s construction also adds the term “operand,” which is not in the claims or

specification and is unnecessary to defining "finite field operation." "Operand" is, however, used in other claims of other patents at issue in this case (*see* '286 Pat., Claim 1), making its inclusion here potentially (and unnecessarily) confusing.

2 & 3. "reduced result" / "unreduced result" *('960 patent – claims 3, 6; '062 patent – claims 1-4, 6, 7)*

Claim Term	Defendant's Constructions	Plaintiffs' Revised Constructions[10]
reduced result	No construction needed. Plain and ordinary meaning.	"result of performing the claimed modular reduction"
unreduced result	"result without any reduction to a specific finite field or wordsize reduction"	"result without performing the claimed modular reduction"

The terms "reduced result" and its antonym, "unreduced result," are both in dispute. Defendant's opening brief addresses its construction for "unreduced result" but does not offer a construction for "reduced result," saying only that it has its "ordinary meaning." Because the claims themselves define what it means to *reduce* a result, and because *unreduced* means the opposite of that, the terms should be construed consistently.

a) "reduced result"

The plain and ordinary meaning of "reduced result," as recited in the claims, is: "result of performing the claimed modular reduction." The claims in both patents make this clear:

> **('960 patent) 3.** A method of performing a finite field operation on elements of a finite field, comprising the steps of:
>
> [a] representing each element as a predetermined number of machine words;
>
> [b] performing a non-reducing wordsized operation on said representations, said wordsized operation corresponding to said finite field operation;
>
> [c] completing said non-reducing wordsized operation for each word of said representations to obtain an unreduced result;

[10] Plaintiffs proposed their revised constructions to Defendant on January 11, 2026, noting that they hoped it would help the parties come to resolution. Defendant notified Plaintiffs on January 12 that it did not agree to either proposal.

[d] upon computing said unreduced result, performing a specific modular reduction of said unreduced result to reduce said unreduced result to that of a field element of said finite field to obtain a ***reduced result***; and

[e] using said ***reduced result*** in a cryptographic operation.

('062 Patent) 1. A method of performing a finite field operation on elements of a finite field, the method comprising a processor:

[a] obtaining a first set of instructions for performing the finite field operation on values representing the elements of the finite field;

[b] executing the first set of instructions to generate an unreduced result completing the finite field operation;

[c] obtaining a second set of instructions for performing a modular reduction for a specific finite field;

[d] executing the second set of instructions on the unreduced result to generate a ***reduced result***; and

[e] providing the ***reduced result*** as an output for use in a cryptographic operation.

The plain meaning of "reduced result" is clear—it's what results from performing or executing the "modular reduction" claimed in each claim. This construction is consistent with, and supported by, the patents' specification, which extensively discusses what's entailed in performing finite field operations, including what it means to "reduce" a result through modular reduction. Ex. 1 ¶¶ 86-89. The specification explains, using a finite field multiplication operation as an example, that after the multiplication is performed, the result of the multiplication is "passed … to the finite field reduction [step]." *Id*. at 8:20-24. A "specific reduction" is then executed that "reduces the result." *Id.* at 8:26-54).

Defendant's "no construction needed - plain and ordinary meaning" position aligns with Plaintiffs'. Defendant states (and Plaintiffs agree) that "[t]he claims explicitly define what a 'reduced result' is" by reciting that a reduced result "is a result generated by "performing a ***modular reduction for a specific finite field***…on the unreduced result." Dkt. 52 at 21 (emphasis in original). Because the claims already make the meaning of the term clear, Defendant posits that

"no construction is needed." *Id.* This would be acceptable but for Defendant's position on the term's antonym, "*un*reduced result." There, Defendant argues for a construction untethered from (and at odds with) the claim language and unsupported by the specification. At best, this inconsistent and illogical positioning will cause confusion; more likely, it will lead to conflicting applications of *reduced* and *unreduced* results. To avoid this, "reduced result" and "unreduced result" should be construed consistently, as discussed below.

b) "unreduced result"

The plain and ordinary meaning of "unreduced result," as set forth in the claims, is: "result without performing the claimed modular reduction." It is simply the inverse of "reduced result." Ex. 1 ¶¶ 86-89. This construction is consistent with the claims' language and the intrinsic record. As explained, and as Defendant agrees, the claim explicitly defines a "reduced result" to be what results when the claim's "modular reduction" is performed. For consistency, "unreduced result" must mean the result on which such modular reduction is not performed. This flows from same logic Defendant employed in discussing "reduced result": The claim already recites how an unreduced result is obtained—for claim 3 of the '960 patent, it is by "completing said non-reducing wordsized operation for each word of said representations to obtain an unreduced result" and for claim 1 of the '062 patent, it is by "executing the [claimed] first set of instructions to generate an unreduced result." As such, "unreduced result" could also be left without further construction. Instead, Defendant pushes for a completely different definition that differs from the claim language, the specification's disclosures, and "reduced result." To maintain parity with "reduced result" and to avoid unnecessary confusion, the plain and ordinary definition of "unreduced result" should be consistent with its antonym, *i.e.*, without performing the claimed modular reduction.

The specification supports the understanding that "reduced" and "unreduced" are what results with and without performing modular reductions. For example, it explains that after a finite

field multiplication is performed, the result of the multiplication (*i.e.*, the unreduced result) is then "passed to the finite field reduction [step]." '960 Pat., 8:26-32. Ex. 1 ¶¶ 89. A "specific reduction" is then executed that "reduces the result" to produce a "reduced result." '960 Pat., 8:46-54.

Notwithstanding its "plain and ordinary" position with respect to "reduced result," Defendant argues for a confusing and incongruent construction for "unreduced result": "result without any reduction to a specific finite field or wordsize reduction." Much is wrong with Defendant's position. First, this construction is circular—Defendant defines "unreduced" as "without any reduction" but declines to define what it means to be "reduced" (resting instead on "plain and ordinary"). Next, the inclusion of "***any*** reduction" limits the scope of the claim improperly – the claim only talks about "reduction" in the context of what happens when the required "modular reduction" is performed, not "any" reduction. And finally, Defendant's inclusion of "any reduction *to a specific finite field or wordsize reduction*" it is both unnecessary and improper. The claim already states what type of reduction is *not* performed on an unreduced result, *i.e.,* the "modular reduction." But Defendant's construction appears to say "reduction to" and then details what's being reduced: "a specific finite field or wordsize reduction." This makes little sense. The claims' plain language compels that the "modular reduction" be made to the "unreduced result," *not* some "specific finite field or wordsize reduction." Read in light of the claim, Defendant's construction would means that "a specific finite field or wordsize reduction" is synonymous with "unreduced result." But nothing in the intrinsic record supports this equivalency. Starting with the claims themselves, claim 1 of the '062 patent uses the term "specific finite field" distinctly from "unreduced result," *i.e.*, they are not synonymous. Claim 3 of the '960 patent does not use the term "specific finite field," but does recite "finite field' – again, distinctly from "unreduced result." "Wordsize reduction" (Defendant's terminology) appears nowhere in

either patent's claims, and "*wordsized operation*," which is present only in the '960 patent claims, is also used distinctly from "unreduced result."

C. The '827 and '370 Patents

"Public key cryptography is based upon the generation of a key pair, one of which is private and the other public[,] that are related by a one way mathematical function." '827 Pat. (Dkt. 52-6), 1:24-34.[11] In cryptosystems using the Elliptic Curve Digital Signature Algorithm (ECDSA), the signer selects a long-term private key "d" (which is secret) and computes a long term public key "Q" that is made available to verifiers (and is public). *Id*. at 2:31-41. For any message "M," the signer can create a signature, which is a pair of integers (r, s), and any verifier can take the message M, the public key Q, and the signature (r, s) and verify whether it was created by the signer. *Id*. at 2:42-48. Typically, a signer would send their public key Q with the message M, or the verifier would look it up. *Id*. at 15:15-26.

To avoid sending or looking up the public key, the inventors devised a technique whereby the public key can be omitted from the message and instead recovered from the signature. *Id*. at 15:27-40. According to this technique, the public key is omitted from the message but is recovered using the signature components (r, s) and other available information by computing $Q=r^{-1}(sR-eG)$. *Id.* Because "one can recover several candidate points Q that could potentially be the public key," the patent also teaches ways "to check that Q is [the sending] correspondent's public key." '827 Pat., 15:27-61. This technique allows for savings on bandwidth and storage, which yields reduced verification times. *Id*.

[11] The '827 and '370 Patents share a common specification; for convenience, citations made to the '827 patent are also applicable to the '370 patent.

1. **"which provides for an accelerated verification of the received signature"** *('370 patent – claim 1)*

Plaintiffs agree this term is not limiting.

2. **"the electronic message omits a public key of a signer"** *('370 patent – claim 1)*

Defendant's Construction	Plaintiffs' Construction
"the electronic message does not include any representation of the public key of the signer"	Plain and ordinary meaning

The phrase "the electronic message omits a public key of a signer," as it is used within the context of claim 1 and the specification in the '370 Patent (Dkt. 52-7), would have been understood by a POSITA to have its plain and ordinary meaning, without need for further construction. Ex. 1 ¶ 92. The claim's plain language (below in relevant part) makes the phrase's meaning clear:

> **1.** A method performed by a hardware processor of a computing device, comprising:
>
> [a] receiving, by a receiver of the computing device and through a network, an electronic message including a signature, ***wherein the electronic message omits a public key of a signer***, and the signature comprises a signature on the electronic message M;
>
> [b] …
>
> [c] recovering, by the hardware processor of the computing device, the **omitted public key of the signer** based on the received first elliptic curve point and the received signature, wherein the public key comprises a second elliptic curve point in an elliptic curve group different from the first elliptic curve point, wherein the elliptic curve group includes the first and second elliptic curve points, wherein the second elliptic curve point comprises an elliptic curve point Q, wherein recovering **the omitted public key of the signer** comprises computing $Q=r^{-1}(sR-eG)$, wherein G comprises a generator of an elliptic curve group that includes the elliptic curve point R and the elliptic curve point Q, and wherein e is a hash value computed from the electronic message M; and
>
> [d] …

'370 Pat., Claim 1. Per the claim's language, the electronic message M does not contain a signer's public key. Ex. 1 ¶ 93. This is corroborated by the specification and prosecution history. '370 Pat., 15:52-57 ("If correspondent 12 **did not send her public key**" with the message M, "it would

be beneficial to be able to recover the public key Q from the signature." (emphasis added)); *see also id.* at 16:18-21; Ex. 6 at 4601 ("enabling recipients to recover public keys **when the public keys are omitted in a message**" (emphasis added)). The only discussion of "omitting" anything from a message in the intrinsic record concerns the signer's public key and nothing else. '370 Pat., 15:52-57, 16:18-21; Ex. 6 at 4601; *see also* Ex. 6 at 4534 ("even when the **public keys** are not included in a message"), *id*. at 4444, 4482 ("enabling recipients to determine **public keys** even when not included in a message").

Defendant's construction requires excluding more than a signer's public key. According to Defendant, not only should the electronic message not include a public key, it also should not include "*any representation* of the public key." This is problematic for several reasons. First, the term "representation of the public key" does not appear in the intrinsic record, and Defendant does not explain what by "any representation of" means or whether or how it alters the scope of the public key itself. Defendant's only example of a purported "representation" of a key is "a compressed version of the public key Q." Dkt. 52 at 25. But the inventors knew how to refer to a "compressed version" of a point (such as Q) when desired. Ex. 1 ¶¶ 94-95; '370 Pat., at 12:15-20 (referring to "compressed version of R"). And the intrinsic evidence consistently refers to omitting a "public key," not a compressed version or other purported "representation" of it. *See supra*; *see also Acumed LLC v. Stryker Corp.*, 483 F.3d 800, 807 (Fed. Cir. 2007) ("perpendicular" and "transverse" had different meanings, and the patentees knew how to say "perpendicular" if that's what they meant). In fact, the specification expressly teaches that a "compact value derived from Q" (or "compact version of Q") may be sent "***instead*** of Q." '370 Pat., 16:22-27; Ex. 1 ¶ 95.

Critically, Defendant's construction would read out every embodiment of the invention. According to Defendant, a "representation of the public key Q" is something that "can be used to

compute the public key Q." Dkt. 52 at 26. If so, even the received signature of claim 1 would have to be omitted because its components (*r* and *s*) are used to compute the public key Q. Ex. 1 ¶ 95. But that would make claim 1 impossible because it requires receiving (not omitting) the signature with the message. '370 Pat., Claim 1 ("receiving … an electronic message ***including a signature***"); *id*. at 15:48-57. *Kaneka*, 790 F.3d at 1304 ("A claim construction that excludes a preferred embodiment is rarely, if ever, correct. A construction that excludes *all* disclosed embodiments … is especially disfavored." (cleaned up)).

Defendant glosses over facts from the prosecution history that undermine its argument. Defendant points to applicants' statement that because a prior art reference included a "short term public key R" in the message, it did not disclose "omitting ***any public keys*** of the signer." Ex. 6 at 4448 (emphasis added). But this argument was made in relation to the as-then-drafted claim, which required the message to omit "***any public key*** of a signer." *Id*. (emphasis added). In other words, applicants did not argue that the reference failed to teach omitting "any public key" because the short term public key in the message could be used to compute the public key Q, as Defendant incorrectly suggests—it was because the short-term public key was an un-omitted "public key" of the signer. Ex. 6 at 4448; Ex. 1 ¶¶ 96-98. Indeed, the claims did not refer to Q then. *See supra*.

In a subsequent amendment, the applicant replaced "***any*** public key" with "***a*** public key" that specifically corresponds to "a second elliptic curve point" comprising a point "Q." *Id*.

1. **(Currently Amended)** A method **performed by a hardware processor of a computing device**, comprising:

receiving, at ~~**a verifier**~~ **the computing device** and through a network, an electronic message including a signature, wherein the electronic message omits **[[any]] a** public key of a signer**, and the signature comprises a signature on the electronic message *M***;

> ~~generating~~recovering, at the ~~verifier and using data processing apparatus~~ computing device, [[a]] the omitted public key of the signer based on the received first elliptic curve point and the received signature, wherein the public key comprises a second elliptic curve point in an elliptic curve group different from the first elliptic curve point, wherein the elliptic curve group includes the first and second elliptic curve points, wherein the second elliptic curve point comprises an elliptic curve point *Q*, wherein recovering the omitted public key of the signer

Ex. 6 at 4595. So, even if the claims previously required omitting other public keys that can be used to calculate Q (what Defendant calls a "representation of the public key Q"), the issued claims do not. Ex. 1 ¶ 98.

3. "verifying that the second elliptic curve point Q represents the public key of the signer" *('827 patent – claim 2)*

Defendant's Construction	Plaintiffs' Construction
"verifying that the second elliptic curve point Q represents the second elliptic curve point Q"	Plain and ordinary meaning

The phrase "verifying that the second elliptic curve point Q represents the public key of the signer," as it is used within the context of the claim and the specification, would have been understood by a POSITA to have it plain and ordinary meaning, without need for further construction. Ex. 1 ¶ 99. Claim 1 recites that the "public key…comprises a second elliptic curve point Q," and that Q is computed from the equation $Q=r^{-1}(sR-eG)$. *See* Claim 1[c] *supra.* Claim 2 then requires verifying that the public key is Q:

> **2.** The method of claim 1, further comprising ***verifying that the second elliptic curve point Q represents the public key of the signer***.

This is consistent with the specification, which teaches that "one can recover several candidate points Q that could potentially be the public key," thus "the [receiving] correspondent 14 needs a way to check that Q is [the sending] correspondent's 12 public key." '827 Pat., 15:27-43. Ex. 1 ¶¶ 99-100.

Defendant characterizes its own construction—which rewrites the claim to require

verifying that "Q is Q"—as "nonsensical," and in fact it is. Ex. 1 ¶ 101. It is also unsupported by anything in the intrinsic record. *Becton, Dickinson & Co. v. Tyco Healthcare Grp., LP*, 616 F.3d 1249, 1255 (Fed. Cir. 2010) ("A claim construction that renders asserted claims facially nonsensical cannot be correct." (cleaned up)).

D. The '961 Patent

Public key cryptosystems can be used to digitally sign messages to authenticate the sender. '961 Pat. (Dkt. 52-8), 1:26-32. The sender signs the message with their private key, and a recipient can verify the message by applying the sender's corresponding public key. *Id*. To be secure, the private key must remain secret, so protocols have been developed that incorporate additional, short-term keys (*e.g.*, ephemeral keys) that are used temporarily. *Id*. at 1:33-50. An ephemeral private key is usually generated by a random number generator ("RNG"), thus it will have a uniform distribution throughout the range of possible values. *Id*. at 1:33-50, 2:15-32. But techniques for generating random numbers (including those that generate a seed value from a RNG and hashing it) can inadvertently introduce a bias that favors certain intervals of key values, which can be exploited to discover private keys, rendering the system insecure. *Id*. at 2:23-61.

The '961 Patent teaches techniques for key generation "in which any bias is eliminated during the selection of the key." *Id*. at 3:1-3. The process includes generating a seed value from a RNG, hashing it, determining whether the output is within an acceptable range, and accepting it if it is, and rejecting it if it's not. *Id*. at 3:64-4:17. If the output is rejected, the method is repeated such that either another seed value is generated by the random number generator or the output is incremented, for example, with a deterministic function. *Id*.; *see id* at 4:18-52.

1. "random number generator" *('961 patent – claims 1-7)*

Defendant's Construction	Plaintiffs' Construction
"a system or algorithm that generates a random value"	"computer instructions capable of generating values according to a uniform random probability distribution"

According to its conventional and well-understood meaning, a "random number generator" generates values according to a uniform random probability distribution. Ex. 1 ¶ 104. Defendant's construction is intended to limit RNGs to only "true" RNGs (a type of RNG that generates truly random values), and exclude other types of RNGs that use "deterministic functions or algorithms" that generate pseudorandom values. Dkt. 52-2, ¶¶ 84, 85. Nothing in the intrinsic record supports limiting the term's ordinary meaning in the manner Defendant wants.

a) The ordinary meaning of "random number generator" includes generation of pseudorandom numbers

The '961 Patent uses "random number generator" consistently with its ordinary meaning, which is not limited to true random number generators. Ex. 1 ¶¶105. The patent explains that an RNG selects values with "a uniform distribution throughout the defined interval" of possible values. '961 Pat., 2:15-17, 2:29-32. The patent consistently refers to "random number generator" in this ordinary sense without ever limiting it to true random number generators or excluding pseudorandom number generators.[12] *See* '961 Pat., 1:41-43, 2:29-32, 2:40-43, 3:33-38, 3:64-66, 4:7-10, 4:13-16, 4:18-23, 4:37-39, 4:50-52, 5:1-4; *Thorner v. Sony Computer Ent. Am. LLC*, 669 F.3d 1362, 1367-68 (Fed. Cir. 2012) (lexicography and disavowal both "require a clear and explicit statement by the patentee"). And it does so regardless of whether it is describing the generation of a key (*e.g.*, k) or a seed value (*e.g.*, SV). *Id.*

[12] The prosecution history also lacks any disclaimer of pseudorandom number generators from the meaning of "random number generator." *Omega Eng'g, Inc, v. Raytek Corp.*, 334 F.3d 1314, 1325-26 (Fed. Cir. 2003) (prosecution history disclaimer must be clear and unmistakable).

That is consistent with the well-known goal of an RNG. Ex. 1 ¶ 106; Ex. 7 at 170 ("A random bit generator can be used to generate (**uniformly distributed**) **random numbers**."), 40 (examples of a number "**generated randomly from a uniform distribution**" using a container of numbered balls); *see also* Ex. 8 at 438 (defining "random number generation" as the "[p]roduction of an unpredictable sequence of numbers in which no number is any more likely to occur … than any other"); Ex. 4 (Koç Dep. Tr.) at 92:8-19 (testifying that random numbers "have to be uniformly distributed over the values of the numbers").

Generating random values according to a uniform distribution can be, and often is, done using pseudorandom number generators ("PRNG"), as confirmed by well-known extrinsic references. Ex. 1 ¶ 107; *see* Ex. 7 at 186 (describing Micali-Schnorr pseudorandom bit generator that relies on values "indistinguishable … from the **uniform distribution** of integers in the interval [0, *n*-1]"); Ex. 8 at 438 ("The process used in computers would be more properly called '**pseudorandom number generation**.'"); Ex. 9 at 172 (defining *random numbers* as "[n]umbers that are drawn from a set of permissible numbers and that have no detectable pattern or bias" and "have an **equal probability of being selected**. … [Computer] programs are designed to generate what are known as **pseudorandom numbers** … [that] are sufficiently random for the purpose intended."). The Handbook of Applied Cryptography[13] says:

> The term ***random numbers***, when used in the context of identification and authentication protocols, **includes pseudorandom numbers** which are unpredictable to an adversary …; this differs from randomness in the traditional statistical sense. In protocol descriptions, **"choose a random number" is usually intended to mean "pick a number with uniform distribution from a specified sample space" or "select from a uniform distribution."**

Ex. 7 at 398 (emphasis added).

[13] The Handbook of Applied Cryptography is a well-known and commonly referenced text. Ex. 1 ¶ 107. Defendant's expert, Dr. Koç, also relies on it.

This is consistent with how RNGs would have been understood by a POSTIA in the context of cryptography. Ex. 1 ¶ 108. A POSITA would understand that RNGs fall into two main categories: (1) "true" RNGs and (2) deterministic (pseudorandom) RNGs. "True" RNGs typically generate values using physical or natural sources of randomness. Ex. 1 ¶ 108; Ex. 7 at 40-41 ("most *true sources* of random sequences … come from *physical means*"), 171 ("A (true) random bit generator requires a naturally occurring source of randomness."), 172 (listing examples of natural and physical sources of randomness for hardware and software generators). Such natural sources can include, for example, time between emissions of particles during radioactive decay, thermal noise, frequency instability of a free running oscillator, capacitor discharge over a fixed time, latency caused by air turbulence in a sealed disk drive, sound from a microphone or video input from a camera. Ex. 7 at 0630. Deterministic RNGs (pseudorandom number generators) generate values using deterministic algorithms (algorithms that produce the same output given the same input). Ex. 1 ¶ 109; *See* Ex. 7 at 41 ("The pseudorandom sequences **appear to be generated by a truly random source** to anyone not knowing the method of generation."), 1708 ("A pseudorandom bit generator (PRBG) is a deterministic algorithm which, given a truly random binary sequence of length k, outputs a binary sequence of length $l >> k$ which **'appears' to be random**."), 171 ("Two general requirements are that the output sequences of a [pseudorandom bit generator] should be statistically **indistinguishable from truly random sequence** …."). Pseudorandom number generators, especially those proven to be "cryptographically secure," generate values considered to be sufficiently "random" for practical purposes. Ex. 1 ¶ 110; Ex. 7 at 170 ("**Cryptographically secure** pseudorandom bit generators are the topic of §5.5."), 399 ("Many protocols involving random numbers require the generation of **cryptographically secure** (i.e., unpredictable) random numbers. If pseudorandom number generators are used, an initial seed

with sufficient entropy is required."); *see also* Ex. 9 at 172 (*random numbers*: "… in practice the **pseudorandom numbers generated by a particular program are sufficiently random for the purpose intended**"). A POSITA would have understood that *true* RNGs can be impractical and would not reasonably be what the invention is limited to. By the same token, a POSITA would have considered PRNGs to be a type of random number generator. Ex. 1 ¶ 110.

A canvasing of specialized computer and computer science dictionaries likewise confirms that RNGs include pseudorandom number generators. Ex. 1 ¶ 111; *Phillips*, 415 F.3d at 1322[14]; Ex. 10 at 310 ("A program that generates a simulated random number (also called a pseudo-random number)."); Ex. 11 at 405 ("[a] software routine that generates a sequence of PSEUDO-RANDOM NUMBERS for use by a larger program…."); Ex. 8 at 438 ("The process used in computers would be more properly called 'pseudorandom number generation.'"); Ex. 9 at 172 ("… In a computer true random numbers are difficult to obtain. Instead programs are designed to generate what are known as pseudorandom numbers….").

Defendant's expert, Dr. Koç, authored a book in which he confirmed that deterministic RNGs (*i.e.*, pseudorandom number generators or PRNGs) *are a type of RNG*, adding to the unbroken corpus of literature on the topic. Ex. 4 at 93:18-23, 100:21-101:2. He also testified that some are used in cryptography. *Id*. at 94:21-95:6 (acknowledging "cryptographically secure PRNG" is an RNG), 104:2-18 (acknowledging that "cryptographically secure RNGs" is a "class of DRNG, which is a [PRNG]"). He further confirmed that deterministic RNGs and true (non-deterministic) RNGs are two classes of RNGs. *Id*. at 101:3-20; Ex. Ex. 12 at 7.

[14] *See also id*. at 1324 ("[A] judge who encounters a claim term while reading a patent might consult a general purpose or specialized dictionary to begin to understand the meaning of the term, before reviewing the remainder of the patent to determine how the patentee has used the term.").

b) Defendant's arguments for narrowing the plain meaning lack support

Defendant offers various arguments in support of its construction, but all are faulty. To begin, Defendant posits (incorrectly) that because PRNGs are deterministic and produce values that are not "actually random," they cannot be *random* number generators. Dkt. 52 at 29-30. But Defendant's emphasis on the word *random* is misplaced. In the context of the '961 Patent, "random number generator" is not a distinction from deterministic (pseudorandom) number generators; it is used in its ordinary sense, which encompasses deterministic RNGs and true RNGs. Ex. 1 ¶¶ 105-111; *Thorner*, 669 F.3d at 1367-68. (declining to construe term narrower than its ordinary meaning where "[t]he specification does not redefine [it] nor is there any disavowal").

Next, Defendant contends claim 7's (which depends from claim 1) use of a "*deterministic* function" for incrementing the output of claim 1's hash function means that the "*random* number generator" in claim 1 cannot be deterministic. This argument relies on the incorrect assumption that "random" must mean *true* random, but this is not the case. Using the word "deterministic" to describe the incrementing function in claim 7 does not change the ordinary meaning of "random number generator" in claim 1 (which encompasses deterministic RNGs). Ex. 1 ¶ 112.

Third, Defendant claims the specification's discussion of the prior art Digital Signature Standard ("DSS") supports its construction. But DSS never says "random number generators" must exclude PRNGs—rather, the term covers both types of random numbers that may be used in DSS, *i.e.*, "[true] random ***or*** pseudorandom integers." Dkt. 52-17 at 2075; Ex. 1 ¶ 113.

Defendant argues that during prosecution, the applicants described the "seed value" generated by the RNGs as a "random" value and uses that to push the argument that, because a seed is generated from an RNG, a *random* seed must mean *true* random. But because the ordinary meaning of "random" includes pseudorandom, the applicants' statement cannot be considered to

disclaim its ordinary scope without an "unambiguous disavowal that clearly and unmistakably disclaims [the] scope or meaning" of the term. *Grober v. Mako Prods., Inc.*, 686 F.3d 1335, 1342 (Fed. Cir. 2012). There was no such disavowal, leaving the full scope of "random" undisturbed.[15]

Finally, Defendant points to alleged prior art references that distinguish between generators of *true* random numbers and pseudorandom numbers. That both *true* and *pseudo* random numbers exist is not in dispute. For reasons stated, the '961 patent uses "random" in its ordinary sense, which encompasses both types of numbers. The fact that some references refer to "true" random simply as "random" does not prove otherwise. *See Boehringer Ingelheim Vetmedica, Inc. v. Schering-Plough Corp.*, 320 F.3d 1339, 1347 (Fed. Cir. 2003) ("a patentee does not renounce the ordinary meaning of a term merely by submitting a reference that employs a different meaning").

c) Case law supports an ordinary meaning of "random" to include pseudorandom, absent disclaimer

Other courts have declined to restrict the meaning of "random" to exclude pseudorandom where, as here, the intrinsic record does not support such a restriction. In *Northeastern University v. Google, Inc.*, No. 2:07-CV-486-CE, 2010 WL 4511010 (E.D. Tex. Nov. 9, 2010), the court recognized "[i]t is generally understood that computers do not perform 'randomly,' but may be programmed to behave pseudorandomly. In other words, 'random' in computer science means that something approximates random, not that it is random." *Id*. at *6. Recognizing there are "different kinds of 'randomness' in computer science," the court declined to construe "randomly selecting" as limited to a "specific kind of random selection," and did not allow the parties "to argue to the jury that the pseudorandom number generators commonly employed as random

[15] Defendant's assertion that the patentee distinguished claim 1 from prior art on the basis that the claimed random number generator generates a random value is misleading and incorrect. None of the patentee's statements cited by Defendant distinguish prior art based on the difference between "random" (*i.e.*, true random) and pseudorandom. In fact, the PTAB found that the difference was "no[t] germane." Ex. 13 at 16 n.7.

number generators in modern computers are not, in fact, random." *Id*. at *7 Similarly, in *California Inst. of Tech. v. Broadcom Ltd.*, No. CV 16-3714-GW (AGRX), 2019 WL 11828243 (C.D. Cal. Nov. 25, 2019), the court considered whether "random permutation" "requires a purely random process of permuting bits." *Id*. at *5. The defendant argued "random" excludes the use of deterministic algorithms, *i.e.*, "where the same input always has the same output." *Id*. at *6. The court rejected that argument because the intrinsic record did not support limiting the meaning of "random" to exclude pseudorandom. *Id*. Defendant's construction and supporting arguments mirror those rejected by these courts and should be similarly rejected.[16]

2. "seed" *('961 patent – claims 1-7)*

Defendant's Construction	Plaintiffs' Construction
"a random value that is used as the starting value for a cryptographic key generation function"	"a value obtained from a random number generator that is used to as the starting value for a cryptographic key generation function"

The parties agree that "seed" is a "starting value for a cryptographic key generation function." The dispute is limited to whether that starting value is restricted to truly random values (as Defendant wants) or may include pseudorandom values (as Plaintiffs proposes). As discussed above, the ordinary meaning of "random" is not limited to true random values but also includes pseudorandom values. "Seed" values are similarly understood to include pseudorandom values.

a) The ordinary meaning of "seed" includes pseudorandom values

The ordinary meaning of "seed" encompasses pseudorandom values. Ex. 1 ¶ 114; Ex. 8 at 471 (*seed*: "A starting value used in generating a sequence of random or pseudorandom numbers."); Ex. 11 at 436 (*seed*: "An initial number supplied to a computer's RANDOM

[16] A construction closer to the term's plain and ordinary meaning (or even a variant of Defendant's proposed construction) might be acceptable if the term "random" therein is understood to include (not exclude) pseudorandom. To that end, a limiting instruction like the one issued in *Northeastern* may be useful here.

NUMBER GENERATOR to begin a new number sequence.").

The '961 Patent does not limit "seed" beyond its conventional meaning. Ex. 1 ¶ 115; '961 pat., 2:40-43; 3:64-66; 4:37-39; 4:49-51; claims 1, 2, 9, 10, 16, 22-24. The patent's claims and specification consistently describe the "seed" as being generated from a "random number generator," which, as explained, is not limited to generators of true random values and encompasses generators of pseudorandom values. *Id*.; § II.D.1, *supra*. Plaintiffs' construction properly captures this scope with the phrase "a value obtained from a random number generator," which indicates that a "seed" is a random value insofar as it is generated by a "random number generator," but without limiting it to a specific type of random value.[17] Defendant's construction, on the other hand, incorrectly limits "seed" to *true* random values[18] (a type of random value) and excludes pseudorandom values (another type of random value).

3. "The method of claim 1 wherein if said output is rejected, said output is incremented by a deterministic function and a hash function is performed on said incremented output to produce a new output; a determination being made as to whether said new output is acceptable as a key."
('961 patent – claim 7)

Defendant's Construction	**Plaintiffs' Construction**
Indefinite	Not indefinite.

Claim 7 is not indefinite. A POSTIA would have readily understood the claim, as written, in view of the intrinsic record.

Claim 7 depends from claim 1. Claims 1 and 7 recite, in relevant part:

[17] The portions of the prosecution history and IPR record Defendant cites similarly reflect that a "seed" is "random" or "randomly" generated insofar as it is generated by a "random number generator"—a proposition that Defendant appears to agree with. Dkt. 52 at 33 (arguing that "the seed value is random because it is generated from a random number generator"). They do not, however, limit a "seed" to a specific type of randomness, *e.g.*, true random or pseudorandom.

[18] As with "random number generator," Defendant limits "random" to *true* random, which it distinguishes from "pseudorandom" and "deterministic."

> **1.** A method of generating a key k for use in a cryptographic function performed over a group of order q, said method including the steps of:
>
> [a] generating a seed value SV from a random number generator;
>
> [b] performing a hash function H( ) on said seed value SV to provide an output H(SV);
>
> [c – d] …
>
> [e] rejecting said output H(SV) as said key if said value is not less than said order q;
>
> [f] if said output H(SV) is rejected, repeating said method; and
>
> [g] …
>
> **7.** The method of claim 1 wherein if said output is rejected, said output is incremented by a deterministic function and a hash function is performed on said incremented output to produce a new output; a determination being made as to whether said new output is acceptable as a key.

Claim 7 only occurs where the "output H(SV)" of Claim 1 is rejected. When this happens, element 1[f] requires that the method of claim 1 be repeated. Dependent claim 7 then adds additional limitations when the method is repeated. Ex. 1 ¶ 118. This is supported by the specification, which teaches different ways of generating a key when an output is rejected. '961 Pat., 4:50-52 ("Upon rejection, the random number generator may generate a new value as disclosed in FIG. 2 or may increment the seed value as disclosed in FIG. 3."). Claim 7 reflects the method of Fig. 3, which teaches the additional step of incrementing the rejected output by a deterministic function before hashing it. Ex. 1 ¶ 118; '961 Pat., Fig. 3; 4:18-31. A POSTIA would have understood claim 1 to encompass multiple ways of performing the same method, and that claim 7 merely claims another variant. Ex. 1 ¶ 118. *BASF Corp. v. Johnson Matthey Inc.*, 875 F.3d 1360, 1367 (Fed. Cir. 2017) ("[B]readth is not indefiniteness.").

Defendant's indefiniteness argument relies on an impermissibly narrow reading of "repeating *said method*." As explained, the '961 patent teaches multiple ways to "repeat" that method when an output is rejected. One way is to have the random number generator generate another seed value, as required in claim 2. '961 Pat., Claim 2 ("wherein another seed value is

generated by said random number generator"); Ex. 1 ¶ 119. Another way involves additionally incrementing the rejected output before hashing it, as required in claim 7. '961 Pat., Claim 7; Ex. 1 ¶ 119. Defendant's argument that claim 7 can only be an "alternative" to claim 1—as opposed to claim 1 with an additional requirement—improperly restricts claim 1 to only one particular way of carrying out the claimed method.[19]

III. CONCLUSION

For the foregoing reasons, Plaintiffs respectfully requests that the Court enter an order finding the aforementioned terms not indefinite, adopting Plaintiffs' proposed constructions, and rejecting Defendant's arguments and proposed constructions.

[19] Defendant's argument that there are multiple ways to interpret the requirements of claim 7 does not mean that the scope of claim 7 lacks reasonable certainty. *ClearOne, Inc. v. Shure Acquisition Holdings, Inc.*, 35 F.4th 1345, 1351 (Fed. Cir. 2022) (indefiniteness is not established by "merely identify[ing] different ways one could interpret" the claim); *Nevro Corp. v. Bos. Sci. Corp.*, 955 F.3d 35, 41 (Fed. Cir. 2020) ("The test is not merely whether a claim is susceptible to differing interpretations.").

Dated: January 14, 2026

Respectfully Submitted,

/s/ Philip J. Eklem
Philip J. Eklem
REICHMAN JORGENSEN LEHMAN & FELDBERG LLP
1909 K Street NW, Suite 800
Washington DC, 20006
Tel: (202) 894-7310
peklem@reichmanjorgensen.com

Khue V. Hoang
REICHMAN JORGENSEN LEHMAN & FELDBERG LLP
400 Madison Avenue, Suite 14D
New York, NY 10017
Tel: (212) 381-1965
khoang@reichmanjorgensen.com

Matthew G. Berkowitz
REICHMAN JORGENSEN LEHMAN & FELDBERG LLP
100 Marine Parkway, Suite 300
Redwood Shores, CA 94065
Tel: (650) 623-1401
mberkowitz@reichmanjorgensen.com

Of Counsel:

Mark D. Siegmund, TX Bar No. 24117055
Cherry Johnson Siegmund James, PC
7901 Fish Pond Rd., 2nd Floor
Waco, TX 76710
Telephone: (254) 732-2242
Facsimile: (866) 627-3509
Email: msiegmund@cjsjlaw.com

Attorneys for Plaintiffs
Malikie Innovations, Ltd. and
Key Patent Innovations, Ltd.

CERTIFICATE OF SERVICE

I hereby certify that all counsel of record are being served with a copy of the foregoing document via the Court's CM/ECF system on January 14, 2026.

/s/ Mark D. Siegmund

Mark D. Siegmund